# 360 语音识别 ANDROID 平台 SDK 说明文档

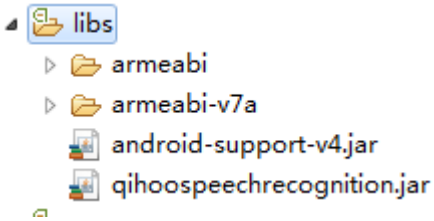
# 一.底层 SDK 使用步骤

1. 语音识别 Android 平台 SDK 使用前需提供相应的 AppId 和 AppSecret；

   系统部提供测试使用 AppId：voice_test　appSecret：CbzUNsiSyt+4mcO44DCvC5Kb1sk=

   对应的 AppId 和 AppSecret 申请地址：voice.360.cn

1. 将 SDK 目录下所有的文件添加到工程的 lib 目录中；




2. 在应用的 AndroidManifest.xml 添加如下权限：

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

3. 在 AndroidManifest.xml 的<application>标签中添加如下代码：

```
<service android:name="com.qihoo.speechrecognition.QihooRecognitionService"/>
```

4. 实现 android.speech.RecognitionListener 回调接口，在此接口对各种语音事件及错误情况进行处理；错误码详见附录一，接口详见附录二；

5. 创建 com.qihoo.speechrecognition.QihooSpeechRecognizer 的实例：

```
QihooSpeechRecognizer speechRecognizer =
QihooSpeechRecognizer.createSpeechRecognizer(getApplicationContext(), appId, appSecret);
```

6. 配置 SDK：

```
QihooSpeechConfig mSpeechConfig = new QihooSpeechConfig();
mSpeechConfig.setmBUseEndPoint(false);   /*是否使用自动检测语音尾点*/
mSpeechConfig.setmConnTimeout(5);        /*wifi 情况下连接超时，默认 5s*/
mSpeechConfig.setmReadTimeout(20);       /*wifi 情况下连接超时，默认 20s*/
mSpeechConfig.setmNonWifiConnTimeout(10);
mSpeechConfig.setmNonWifiReadTimeout(30);
mSpeechRecognizer.setConfig(mSpeechConfig);
```

   请结合实际情况设置适当的参数。

7. 为步骤 6 创建的 speechRecognizer 设置 Listener：

```
RecognitionListener listener = new YourRecognitionListener;
speechRecognizer.setRecognitionListener(listener);
```

8. 调用 startListenering 开始录音：

```
speechRecognizer.startListening();
```

9. 目前 SDK 提供自动检测语音尾点功能，停止说话后，自动进行识别。但在噪声情况下，尾点检测有能失效，因此需要主动停止录音：

```
speechRecognizer.stopListening();
```

10. 使用完成后，调用 speechRecognizer.destroy 销毁创建的 QihooSpeechRecognizer 实例。

```
speechRecognizer.destroy();
```

# 二.注意事项

1. 传统模式下，调用 startListener 后，手机处于录音状态，如果不调用 stopListener 或者 cancel 方法，则将一直处于录音状态，其他应用无法获取录音机进行录音。因此，在界面退出等操作中，望业务方调用相应的释放资源方法。

2. 尾点检测模式下，并非所有环境都能正确检测到语音尾点，因此在使用尾点检测模式，希望业务方能够给用户提供在录音时手动终止录音的操作。其他公司的包含尾点检测算法的语音录入功能产品都包含此功能。即如下图上所示。


在录音的同时，提供取消、手动终止录音操作。如果取消，则不返回识别结果。

3.代码混淆。如下类不能混淆：

com.qihoo.speechrecognition.QihooSpeechRecognizer

com.qihoo.speechrecognition.QihooSpeechConfig

com.qihoo.speech.proccess.DataProccessor

# 附录一.回调函数 OnError 错误码说明

```
/** 网络操作超时. */
public static final int ERROR_NETWORK_TIMEOUT = 1;

/** 其他网络相关的错误. */
public static final int ERROR_NETWORK = 2;

/** 音频录音错误. */
public static final int ERROR_AUDIO = 3;

/** 服务器错误. */
public static final int ERROR_SERVER = 4;

/** 其他客户端错误. 客户端调用 QihooSpeechRecognizer 中函数顺序错误等，例如：客户端在没有调用 startListening()便调用 stopListening()情况。*/
public static final int ERROR_CLIENT = 5;

/** 没有语音输入 */
public static final int ERROR_SPEECH_TIMEOUT = 6;

/** 没有匹配的识别结果. */
public static final int ERROR_NO_MATCH = 7;

/** 识别器忙. */
public static final int ERROR_RECOGNIZER_BUSY = 8;

/** 权限不足 */
public static final int ERROR_INSUFFICIENT_PERMISSIONS = 9;
```

# 附录二.android.speech.RecognitionListener 回调接口说明

```
private RecognitionListener mListener = new RecognitionListener() {
      /*识别结果返回回调函数，识别完成后，识别结果存储在 results 中*/
      @Override
      public void onResults(Bundle results) {
          //获得识别结果及 RequestID
          ArrayList<String> strList =
      results.getStringArrayList(QihooSpeechRecognizer.RESULTS_RECOGNITION);

          String reqId =
            results.getString(QihooSpeechRecognizer.RESULTS_REQUESTID);

      }

      /*语音识别过程中出现错误将回调此函数，具体错误码参照附录一*/
      @Override
      public void onError(int error) {

      }

      @Override
      public void onReadyForSpeech(Bundle params) {

      }

       /* 结束录音回调函数
        * 当尾点检测算法检测到语音已结束，回调此函数，告知已检测到语音尾点。
        * */
      @Override
      public void onEndOfSpeech() {   /**/

      }


      @Override
      public void onBeginningOfSpeech() {

      }

      @Override
      public void onBufferReceived(byte[] buffer) {
          // TODO Auto-generated method stub

      }

      @Override
      public void onEvent(int eventType, Bundle params) {
           // TODO Auto-generated method stub
      }

      @Override
      public void onRmsChanged(float arg0) {
          // TODO Auto-generated method stub

      }

      @Override
      public void onPartialResults(Bundle partialResults) {

      }
  };
```